Description of the UTBots Robot Soccer Team

Vinicius Zaramella, João P. D. de Oliveira, Vinicio Pereira, Matheus Yudi,
Roni Banazewski, Jean M. Simão and João A. Fabro

Abstract— **This paper presents the development of the algorithms to control the UTBots soccer robot team, used to compete in the Small Size (F-180) category of Robocup. A computer vision algorithm was developed to locate the robots and the ball on the field. A new fuzzy inference system [1], based on the Notification Oriented Paradigm [2], was developed to efficiently evaluate a set of fuzzy rules controlling the behavior of the robots.**

I. Introduction

In this paper, the main problems were addressed, in order to control a team of 5 robots of the F-180 small size Robocup category. The main proposal is to use fuzzy logic to implement a set of behaviors, based on information provided by a computer vision system. Fuzzy logic is then used to calculate the actions of each robot, based on its role on the team (goalkeeper, defender or striker), its position on the field, and the position of the ball. Due to the relatively large amount of fuzzy inference rules (about 42 for each robot, giving a total of at 210), and the necessity of speed for real-time control of the robots, that depending on the camera can be up to 30 times a second, an efficient fuzzy inference engine was developed for this project [1], based on the Notification Oriented Paradigm (NOP), proposed by [2] and detailed in [3].

A NOP framework was implemented, allowing the creation of efficient non-fuzzy rule-based systems. This efficiency is attained by avoiding the need for iterations by all rules, assessing whether the rule should or should not be activated [3]. Using NOP, the attributes of the elements assessed (robots in this case) are reactive: they notify the interested rules of any change in their status, which take the form of reactive agents. So there is no need to iterate through the rule-base to obtain find the active rules. In a previous development of this project, the NOP framework was extended to cope with fuzzy attributes and fuzzy inference, using max-min operators and Mandani-based fuzzy rules, and Average-of-Max defuzification [1].

This work was supported in part by Fundação Araucária, through a PIBIC scholarship granted to V. Zaramella, who is an undergraduate student of Computer Engineering at the Informatics Department (DAINF) of the Technological University of Paraná (UTFPR); e-mail: vinicius_zaramella@yahoo.com.br.

J. P. D. Oliveira is an undergraduate student of Computer Engineering at the Informatics Department (DAINF) at UTFPR; e-mail: joaooliveira.mlp@gmail.com.

V. Pereira is an undergraduate student of Electric Engineering / Automation at the Electronics Department (DAELN) at UTFPR; e-mail:viniciop28@yahoo.com.br.

M. Yudi is an undergraduate student of Electronic Engineering at the DAELN at UTFPR; e-mail:matheusyudits@gmail.com.

R. Banazewski is a PhD student at the CPGEI (Graduate School of Electrical Engineering and Computer Science) at UTFPR. e-mail: ronifabio@gmail.com.

J. Simão is with DAELN and CPGEI ; e-mail:jeansimao@utfpr.edu.br.

J. Fabro is with DAINF and PPGCA Graduation Program on Applied Computing; e-mail: fabro@utfpr.edu.br, (corresponding author: phone: +55(41)3310-4746).

II. The UTBots Team

The UTFPR Robot soccer team is presented in Fig. 1.

Figure 1 - The UTBots Robot Soccer Team

The robots have been acquired [4] by UTFPR as a workbench of study and development for the Computer Engineering undergraduate course. There are several projects associated to the Robot Soccer initiative, including the construction of a new team, and intelligent control using artificial intelligence techniques. The Small Size category operates based on information provided from two top-mounted cameras that provide to the teams images of the field. In 2010, the official rules propose that the organization install the cameras and each team access the images using the SSL open source library [5].

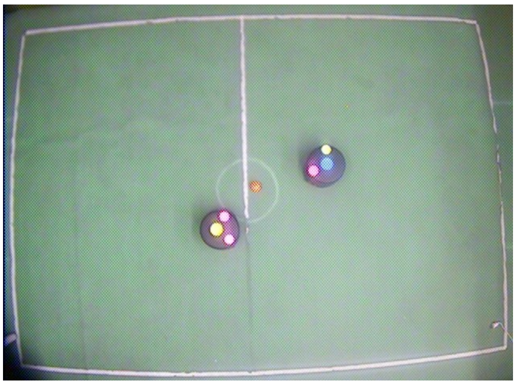

Figure 2- Image of the Field with Robots and the Ball

In our experiments, we used a 3x4 meters field that is visualized on a single frame from a top-mounted camera positioned above the field (Fig. 2).

III. Fuzzy Inference Engine

The Fuzzy Inference Engine (FES) [1] developed for this project is based on the Notification Oriented Paradigm (NOP) [2][3]. In fact, NOP was extended to support fuzzy concepts, and this extension was called NOP-Fuzzy [1].

A. Notification Oriented Paradigm (NOP)

The Notification Oriented Paradigm (NOP) presents a new concept to conceive and execute applications. NOP is based upon the concept of small, smart, and decoupled entities that collaborate by punctual notifications to carry out the software inference execution. A main feature of this inference execution is the avoidance of searches by data, what results in better performance to programs based on NOP in relation to others based on current paradigms in scenarios with large amount of causal expressions [3].

In the current paradigms (i.e. Imperative and Declarative Paradigm), the evaluation process is based upon a monolithic execution or inference system that performs searches over passive fact-bases (e.g. variables and vector sets) and causal-bases (e.g. if-then sets), which generates a set of deficiencies. Precise examples of these deficiencies are the misuse of computation capacity and the coupling of code-parts that respectively generate the degradation of the performance and the hardness to distribute programs.

In NOP, the causal expressions are represented by common causal rules, which is natural to programmers of current paradigms. However, each causal rule is technically dealt with a special computational-entity called Rule. Structurally, a Rule has a Condition and an Action. Both are entities that work together to carry out the causal knowledge of the Rule. The Condition concerns to its decisional part related to the referenced elements, whereas the Action concerns to execution related to these elements.

In NOP, the evaluated elements are represented by a special-entity type called Fact_Base_Element. A Fact_Base_Element (FBE) is composed of one or more attributes. Each attribute is represented by another special-entity type called Attribute.

The Attributes states are analyzable in the Conditions of Rules by means of other collaborator special-entities called Premises. When each Premise of a Rule Condition is in true state, that is concluded by means of a given inference process, the Rule becomes true and can activate its Action which is composed of special-entities called Instigations.

In fact, Instigations are linked to and instigate the execution of Methods, which are another special-entity of Fact_Base_Element. Each Method allows executing services of its Fact_Base_Element. Generally, the call of a Fact_Base_Element Method normally changes one or more Fact_Base_Element Attribute states, thereby feeding the inference process.

The inference process of NOP is innovative once the Rules have their inference carried out by active collaboration of its notifier entities. In short, the collaboration happens in the following way: for each change in an Attribute state of a Fact_Base_Element, the state evaluation occurs only in the related Premises and then only in related and pertinent Conditions of Rules by means of punctual notifications between the collaborators.

B. Fuzzy Extension of NOP (NOP-Fuzzy)

The inference mechanism of NOP was extended to create a fuzzy inference mechanism. Fuzzy systems were proposed by Lotfi Zadeh [6] to deal with imprecise concepts. Zadeh combined the concepts of classical logic with the idea of multivalued logic and proposed treatment of the pertinence to a given set with degrees that may vary in the real interval 0-1.In order to deal with this new logic were developed several computing tools. Systems that could work with the parameters of the new logic and thus creating a rule base to be able to perform fuzzy inferences.
The fuzzy inference engine is composed of a number of nebulous entries that are evaluated by a rule base to generate a nebulous output to be defuzzified, is that, to transform an output fuzzy set in a CRISP value (value with classical logic features).

Fuzzy Rules-Based Systems are composed of "fuzzy rules", such as: "if distance is close and the speed is medium, then the action to take is to brake a lot." Each rule relates to fuzzy linguistic variables in a simple manner, which allows the modeling of human knowledge about the problem even in the presence of imprecision.

Fuzzy logic is a generalization of classical logic. The generalization allows partial matching (fuzzy) between data input and the antecedents of a rule. So when there is a level of activation in all the antecedents of a fuzzy rule, this rule may be "fired", and the activation of its action can be calculated [1]. Thus, the firing of a fuzzy rule contributes only a part of the conclusion. This is because other rules, which have the same linguistic variable in its conclusion, may also contribute to a fuzzy conclusion.

Thus, activation of only one premise will not result in a complete conclusion. It is necessary that all rules that can contribute to fuzzy conclusion to be fired before this conclusion can be properly evaluated and propagated through the system of inference. This problem is usually solved by trying to fire all rules that have the same linguistic variable in the action, even though usually only few rules are active simultaneously. To make this process more effective, the NOP mechanism was used.

IV. PLAYER'S BEHAVIOR

For each robot, its behavior is defined by a set of fuzzy rules that control the action of the robot in the next sense-act cycle. Each sense-act cycle starts with the execution of a computer vision algorithm that receives an image of the field, and discovers the positions of robots and ball. This information is then presented as inputs to the Fuzzy Inference Engine. Each robot has its own rule-base, that bases its decisions on several input variables, such as: its position on the field, its selected role in the team (goalkeeper, defender, or striker), its distance to the ball, and other information that can be obtained from the data presented by the vision algorithms. The following behaviors have been developed for the UTBots team:

A. Striker's Behavior

The strikers were developed based on two different behaviors: to follow the ball and to kick for the goal. The “follow ball” behavior was implemented using the distance between the player and the ball, and its relative angle. Fig. 3 presents the fuzzy sets that encode these input variables. On table 2, the fuzzy rules that encode this behavior are presented. There are two rule-bases, one to define the change in direction that each robot must make to direct itself to the ball, and the second rule-base defines the velocity of the robot. Fig. 4 presents the output variables used in this rules.

Unless the striker robot is very close to the ball, this is its behavior. If there is another player blocking the ball's path, the striker enables its dribbler mechanism, and changes its position randomly. This behavior isn't coded as fuzzy rules, but as a simple algorithm that must be executed by the striker. In simulations, it works, but when tried with the real robot, usually the ball “escapes” once the robots make its turn, and then it returns to the “follow ball” behavior.

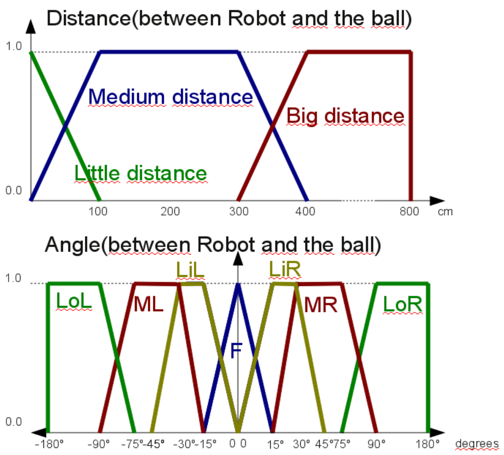

Figure 3 – Fuzzy Input Variables Distance and Angle

Angle → Distance↓	A Lot Left (LoL)	Medium Left (ML)	Little Left (LiL)	Front (F)	Little Right (LiR)	Medium Right (MR)	A Lot Right (LoR)
Little	LS/ TLoL	MS/ TLiL	LS/ TLiL	MS/ F	LS/ TLiR	MS/ TLiR	LS/ TLoR
Medium	LS/ TLiL	MS/ TLoL	MS/ TLiL	HS/ F	MS/ TLiR	MS/ TLoR	LS/ TLiR
Big	LS/ TLiL	MS/ TLoL	HS/ TLiL	HS/ F	HS/ TLiR	MS/ TLoR	LS/ TLiR

Table 2 – Fuzzy Rule-Base for Speed and Rotation determination – Follow Ball Behavior (LS=Little Speed, MS=Medium Speed, HS = High Speed, TLoL = turn a lot to the left, TLiL = turn a little to the left, F = front, TLiR = turn a little to the right, TLoR, turn a lot to the right)

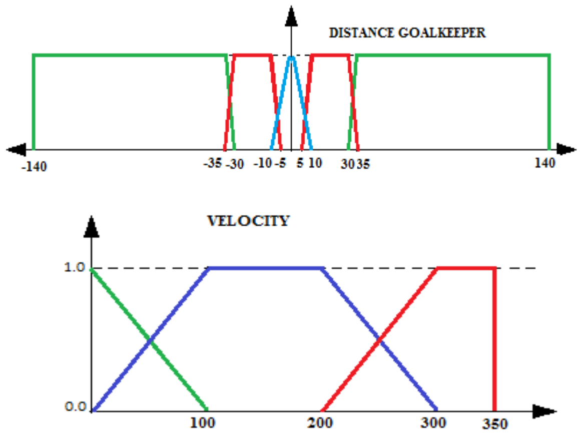

Figure 4 - Fuzzy Output Variables Turn and Velocity

B. Goalkeeper and Defender Behavior

Both the goalkeeper and the defender use similar approaches to define their actions on the field. At first, each robot obtains from the vision system its own position and orientation, and the ball position.

The goalkeeper rules are developed to position this player over a virtual vertical line that is positioned at 25 cm from the goal line of its own field (presented in Fig. 5).

As inputs for the rules, the following variables are calculated: extending a straight line from the ball position to the center of the goal, and intercepting this line with the virtual vertical line where the goalkeeper position itself, it is calculated the distance between the actual coordinate of the goalkeeper and the coordinate of this intersection, giving an positive or negative number, depending on the positions of the ball and the goalkeeper. This value is the input for the rule-base for the goalkeeper. As output, the rule-base decides if the goalkeeper should move to the right or left, and with which velocity, in order to position itself to block a possible kick to its goal. In Fig. 6 are presented the fuzzy sets that define the input (distance from robot to its desired intercepting position) and output (velocity) variables used to implement the behavior of the goalkeeper. Table 3 presents the fuzzy rules implemented.

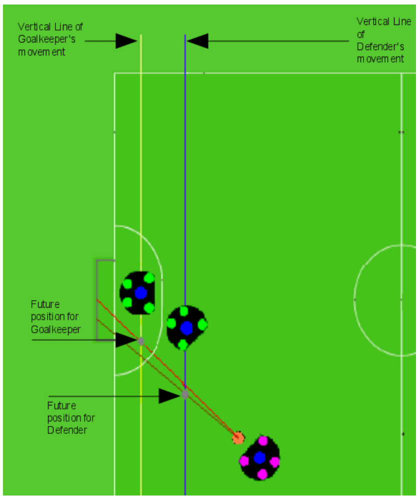

Figure 5 - Goalkeeper and Defender positions

Distance →	Far to the left(FL)	A little to left(LL)	Small (S)	A little to right(LR)	Far to the right (FR)
Velocity →	Fast to left (FL)	Slow to left(SL)	Stop(ST)	Slow to right(SR)	Fast to right(FR)

Table 3: Fuzzy Rule-base for the Goalkeeper

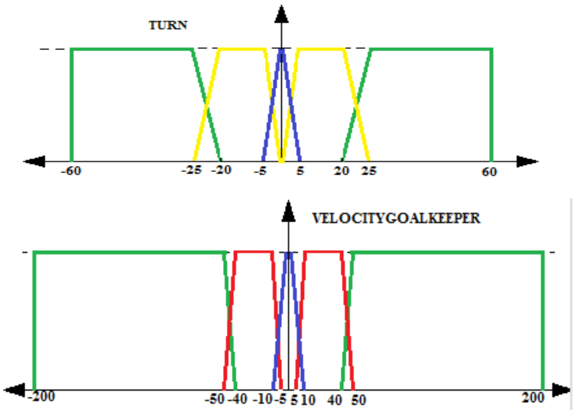

Figure 6 - Input and Output Fuzzy Variables for the Goalkeeper (and Defender) Behavior

As for the defender, the rule-base is similar. There are two robots that are designed defenders, one to the left and one to the right of the defense field. Each of these two robots has its own rules, which use as input the distance that each robot has to move to intercept the ball. But in the case of the defenders, the position to be covered is placed at the back of the goal, but adjusted to the right or to the left. This way, the right defender put itself to "cover" the right corner of the goal, while the left defender covers the left corner of the goal. These three players compose the defense of the UTBots team.

V. Experimental Results

Experiments were realized to verify the quality of the results of the proposed algorithms. The execution of a complete evaluation cycle of fuzzy behavior for all 5 robots took around 200µs (microseconds).

VI. Conclusions and Future Work

In the development of the algorithms for the UTBots robot soccer team, several problems have been found due to low quality of the acquired image, high dependence of illumination conditions, and difficulties when controlling the real robots using radio-frequency. The vision system is too slow and faulty, providing wrong or no robot and ball coordinates to the fuzzy control system. The commands sent to the real robots aren't always correctly executed, and constant re-evaluation of the robot's behavior is necessary. These requirements lead to the use of a simulator [7] during the development of the control algorithms. The simulator has allowed the implementation and debugging of the various fuzzy rule-bases, but when controlling the real robots, new adjustments were necessary, to cope with the greater difficulties presented by the real robots.

Future work includes the development of a second level of fuzzy control rules, to include strategic information, such as ball possession [8] and attacking strategy of the adversary team [9].

References

[1] J. T de Souza, J. A. Fabro, R. F. Banaszewski, J. M. Simão (2009), "Proposta de uma Máquina de Inferência Fuzzy utilizando o Paradigma Orientado a Notificações (PON)", *Proceedings of the 4th C3N – Tri-national Conference*, Oct. 7-9, 2009, Foz do Iguaçu, PR, Brazil (in portuguese).

[2] Simão, J. M. and Stadzisz, P. C. (2009) "Inference Based on Notifications: A Holonic Meta-Model Applied to Control Issues". *IEEE Transactions on Systems, Man and Cybernetics, Part A*. Vol. 39, Issue 1, Jan. 2009 Pg. 238-250.

[3] Banaszewski, R. F. (2009) "Paradigma orientado a notificações: avanços e comparações". Master Thesis, Universidade Tecnológica Federal do Paraná - UTFPR, Graduate School of Electrical Engineering and Computer Science - CPGEI, Curitiba/PR.(in portuguese).

[4] X-Bot SCISoccer, [Online]. Avalilable: http://www.xbot.com.br/produtosss.html.

[5] SSL (Small Size League Shared Vision System) [Online]. Avalilable: http://code.google.com/p/ssl-vision/.

[6] L. A. Zadeh (1965), "Fuzzy Sets" Information and Control, June, 1965, pp. 338-353.

[7] Tewnta F-180 Robocup Simulator [Online]. Avalilable: http://code.google.com/p/tewnta/.

[8] J. R. Neri, C. H. F. Santos, J. A. Fabro (2009)."Sistema especialista fuzzy para posicionamento dos jogadores aplicado ao futebol de robôs". *Proceedings of the IX SBAI- Simpósio Brasileiro de Automação Inteligente*, 2009, Brasília.(in portuguese).

[9] E. A. Maravillas and E. P. Dadios (2010). "FIRA Mirosot Robot Soccer System Using Fuzzy Logic Algorithms", Robot Soccer, V. Papić (Ed.), [Online] Available: http://sciyo.com/articles/show/title/fira-mirosot-robot-soccer-system-using-fuzzy-logic-algorithms.